IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	)	
	)	
Inventors: Guogen Zhang et al.	)	Examiner: Charles D. Adams
	)	
Serial #: 10/629,459	)	Group Art Unit: 2164
	)	
Filed: July 29, 2003	)	Appeal No.: _________
	)	
Title: DYNAMIC SELECTION OF OPTIMAL GROUPING SEQUENCE AT RUNTIME FOR GROUPING SETS, ROLLUP AND CUBE OPERATIONS IN SQL QUERY PROCESSING	)	

BRIEF OF APPELLANTS

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In accordance with 37 C.F.R. §41.37, Appellants' attorney hereby submits the Brief of Appellants on appeal from the final rejection in the above-identified application as set forth in the Office Action dated July 18, 2008.

Please charge the amount of $540.00 to cover the required fee for filing this Appeal Brief as set forth under 37 C.F.R. §41.37(a)(2) and 37 C.F.R. §41.20(b)(2) to Deposit Account 09-0460 of IBM Corporation, the assignee of the present invention. In addition, the Office is authorized to charge any necessary fees or credit any overpayments to Deposit Account No. 09-0460.

I. REAL PARTY IN INTEREST

The real party in interest is IBM Corporation, the assignee of the present application.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences for the above-referenced patent application.

III. STATUS OF CLAIMS

Claims 1-3 are pending in the application.

Claims 4-9 have been canceled from the application.

Claims 1-9 were rejected under 35 U.S.C. §103(a) as being unpatentable over Cheng et al., "Implementation of Two Semantic Query Optimization Techniques in DB2 Universal Database," (Cheng) in view of Cochrane et al., U.S. Patent No. 5,963,936 (Cochrane) and further in view of Al-omari et al., U.S. Patent No. 6,438,741 (Al-omari).

Claims 1-3 are being appealed.

IV. STATUS OF AMENDMENTS

Claims 4-9 were canceled subsequent to the Notice of Appeal in an Amendment under 37 CFR §41.33 submitted on December 18, 2008.

V. SUMMARY OF THE INVENTION

The claimed subject matter is summarized as follows:

Claim element/phrase	Specification Support
1. A method of optimizing a query in a computer system, the query being performed by the computer system to retrieve data from a database stored on the computer system, the method comprising:	Page 3, lines 5-9; Page 6, lines 17-29 referring to reference numbers 200-206 in FIG. 2; Page 28, lines 12-15 referring to FIG. 6.
(a) during compilation of the query, maintaining a GROUP BY clause with one or more GROUPING SETS, ROLLUP or CUBE operations in its original form,	Page 3, lines 9-11; Page 4, lines 19-21; and Page 28, lines 16-18 referring to reference number 600 in FIG. 6.

Claim element/phrase	Specification Support
instead of rewriting the GROUP BY clause, until after query rewrite;	
(b) at a later stage of query compilation, translating the GROUP BY clause with the GROUPING SETS, ROLLUP or CUBE operations into a plurality of levels, wherein each of the levels has one or more grouping sets comprised of grouping columns, and generating a query execution plan for the query with a super group block having an array of pointers, wherein each pointer points to the grouping sets for a particular one of the levels; and	Page 3, lines 12-13; Page 4, lines 21-23; Page 22, lines 10-15 referring to FIG. 4; Page 23, line 18 – page 26, line 2 referring to reference numbers 510, 516 and 518 in FIG. 5; and Page 28, lines 19-22 referring to reference number 602 in FIG. 6.
(c) performing the query execution plan to retrieve data from a database stored on the computer system.	Page 3, lines 5-9; Page 6, lines 28-29 referring to reference number 206 in FIG. 2; Page 28, lines 12-15 referring to FIG. 6.
2. The method of claim 1, further comprising: (1) at query execution time, dynamically determining a grouping sets sequence for the GROUP BY clause with the GROUPING SETS, ROLLUP or CUBE operations based on intermediate grouping sets, in order to optimize the grouping sets sequence.	Page 3, lines 13-16; Page 4, lines 23-26; Page 28, lines 23-29 referring to reference number 604 in FIG. 6.
3. The method of claim 2, wherein the	Page 3, lines 16-18;

Claim element/phrase	Specification Support
dynamically determining step further comprises (1) performing a GROUP BY for a base grouping set and then optimizing execution of the grouping sets sequence by selecting a grouping set having lowest cardinality from a previous one of the levels as an input to a grouping set on a next one of the levels, and (2) performing a UNION ALL operation on the grouping sets.	Page 4, lines 26-28; Page 28, lines 23-29 referring to reference number 604 in FIG. 6.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 1-3 are obvious under 35 U.S.C. §103(a) over Cheng et al., "Implementation of Two Semantic Query Optimization Techniques in DB2 Universal Database," (Cheng) in view of Cochrane et al., U.S. Patent No. 5,963,936 (Cochrane) and further in view of Al-omari et al., U.S. Patent No. 6,438,741 (Al-omari).

VII. ARGUMENTS

A. The Office Action Rejections

In sections (4)-(6) of the Office Action, claims 1-9 were rejected under 35 U.S.C. §103(a) as being obvious in view of the combination of Cheng et al., "Implementation of Two Semantic Query Optimization Techniques in DB2 Universal Database" (Cheng), Cochrane et al., U.S. Patent 5,963,936 (Cochrane) and Al-omari et al., U.S. Patent 6,438,741 (Al-omari).

Appellants' attorney respectfully traverses these rejections.

B. The Appellants' Independent Claim

Independent claim 1 recites a method of optimizing a query in a computer system, the query being performed by the computer system to retrieve data from a database stored on the computer system, the method comprising: (a) during compilation of the query, maintaining a GROUP BY clause with one or more GROUPING SETS, ROLLUP or CUBE operations in its original form, instead of rewriting the GROUP BY clause, until after query rewrite; (b) at a later stage of query compilation, translating the GROUP BY clause with the GROUPING SETS, ROLLUP or CUBE operations into a plurality of levels, wherein each of the levels has one or more grouping sets comprised of grouping columns, and generating a query execution plan for the query with a super group block having an array of pointers, wherein each pointer points to the grouping sets for a particular one of the levels; and (c) performing the query execution plan to retrieve data from a database stored on the computer system.

C. The Cheng Reference

Cheng describes the implementation of two semantic query optimization (SQO) techniques in IBM's DB2 Universal Database. SQO uses integrity constraints associated with a database to improve the efficiency of query evaluation.

D. The Cochrane Reference

Cochrane describes a method and apparatus for detecting and stacking grouping sets to support GROUP BY operations with GROUPING SETS, ROLLUP and CUBE extensions in relational database management systems, with greatly reduced numbers of grouping sets. A first GROUP BY (element-list1) is input to a second GROUP BY (element-list2), resulting in the GROUP BY of the intersection of the two lists. This intersection property is then useable to reduce the number of GROUP BYs required to implement the grouping by GROUPING SETS, ROLLUPs, and CUBEs required for the online analytical processing of data contained in the database.

E. The Al-omari Reference

Al-omari describes a system and method for eliminating compile time explosion in a top down rule based system using selective sampling. The system and method reduces the compile time in a top-down rule based system by identifying the complexity of a query prior to applying a rule to an expression. If the complexity of the query is above a threshold, the present invention determines whether the rule should be applied based upon several factors including the type of rule and the position of the node in the search space. Those rules that need not be applied are randomly pruned at a determined rate that prevents search space explosion and prevents the elimination of large contiguous portions of the search space. Pruned rules are not applied, while those rules that are not pruned are applied.

F. Arguments directed to the first grounds for rejection: Whether claims 1-3 are obvious under 35 U.S.C. §103(a) over Cheng et al., "Implementation of Two Semantic Query Optimization Techniques in DB2 Universal Database," in view of U.S. Patent No. 5,963,936 (Cochrane) and further in view of U.S. Patent No. 6,438,741 (Al-omari).

1. Independent Claim 1

Appellants' invention, as recited in independent claim 1, is patentable over the combination of Cheng, Cochrane and Al-omari, because the claim recites limitations not found in the references.

Nonetheless, the Office Action states the following:

> 3. Claims 1-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cheng et al. ("Implementation of Two Semantic Query Optimization Techniques in DB2 Universal Database"), in view of Cochrane et al. (US Patent 5,963,936), and further in view of Al-omari et al. (US Patent 6,438,741).
>
> As to claim 1, Cheng et al. teaches a method of optimizing a query in a computer system, the query being performed by the computer system to retrieve data from a database stored on the computer system (see Abstract), the method comprising:
>
> (a) during compilation of the query, maintaining a GROUP BY clause (see Cheng et al. Page 1, Example 1, and Page 5, query 1)

Cheng et al. does not teach with one or more GROUPING SETS, ROLLUP or CUBE operations

Cocharane et al. teaches with one or more GROUPING SETS, ROLLUP or CUBE operations (see column 7, lines 26-30, and column 7, lines 44-48)

Cheng et al. as modified teaches in its original form, instead of rewriting the GROUP BY clause, until after query rewrite (see Cheng et al. Page 1, Example 1, and Page 5, query 1. In Q'1, the group by clause has been retained); and

(b) at a later stage of query compilation, translating the GROUP BY clause with the GROUPING SETS, ROLLUP, or CUBE operations into a plurality of levels, wherein each of the levels has one or more grouping sets (see Cochrane et al. 8:26-42, Figure 7. This step occurs after the step listed above) comprised of grouping columns (see 11:62-12:15. The GROUP BY sets are comprised of columns a, b, x, and y),

Cheng et al. as modified does not teach generating a query execution plan for the query with a super group block having an array of grouping sets, wherein each pointer points to the grouping sets for a particular one of the levels.

Al-omari et al. teaches generating a query execution plan for the query with a super group block having an array of grouping sets, wherein each pointer points to the grouping sets for a particular one of the levels (see Figure 3D, 'link mode to GROUP'. Also see 10:36-48, 14:28-35, 41-43)

Cheng et al. as modified teaches:

(c) performing the query execution plan to retrieve data from a database stored on the computer system (see Cochrane et al. 7:41-43).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified Cheng et al. by the teachings of Cochrane et al., since Cochrane et al. teaches that "a method for detecting and stacking grouping sets to support group by operations with grouping sets, rollup, and cube extensions in relational database management systems, with greatly reduced numbers of grouping sets" (see Abstract).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have further modified Cheng et al. by the teachings of Al-omari et al., since Al-omari et al. teaches "a system and method for optimizing complex SQL database queries" (see 3:18-19).

In addition, the Office Action states the following:

Response to Arguments

4. Applicant's arguments filed 28 April 2008 have been fully considered but they are not persuasive.

Applicant argues, in regards to the independent claims, that "Cheng merely shows the GROUP BY clause in the same form in both the original query and the optimized query, indicating that the GROUP BY clause is not 'maintained

during compilation' and then ;translated at a later stage of query compilation,' as recited in Applicants' claims. Instead, the GROUP BY clause of Cheng is apparently left untouched during the join elimination optimization". In response to this argument, it is noted that Cheng et al. maintains the GROUP BY clause between Q1 and Q1'. Thus, the GROUP BY clause is being maintained and not rewritten until the query rewrite goes through this optimization phase of Cheng et al. It is noted that the query is rewritten by the combination of Cheng et al. and Cochrane et al., at a later stage of query compilation (see rejection above).

Applicant argues that "this optimization scheme of Cochrane says nothing about maintaining a GROUP BY clause with one or more GROUPING SETS, ROLLUP or CUBE operations in its original form, instead of rewriting the GROUP BY clause, until after query rewrite". It is noted that the combination of Cheng et al. and Cochrane et al. is used to teach this feature. Cheng et al. teaches a method of optimizing the joins of a query, and Cochrane et al. teaches a method of optimizing the grouping sets of a query into a query graph model.

Applicant argues that "however, the groups from Al-omari are in no way equivalent to Applicants' claimed super group block. Specifically, the memo structure of Al-omari includes one or more groups, where each group contains an array of pointers to one or more logical expressions, an array of pointers to one or more physical expressions, an array of pointers to one or more contexts, an array of pointers to one or more plans, and an exploration pass indicator. In Applicants' claims, on the other hand, the super group block supports the translation of a GROUP BY clause with the GROUPING SETS, ROLLUP, or CUBE operations into the plurality of levels, wherein each of the levels has one or more grouping sets comprised of grouping columns, the super group block has an array of pointers, and each pointer of the super group block points to the grouping sets for a particular one of a plurality of levels. This super group block of Applicants' recites different structure and functions as compared to the memo structure of Al-omari". Examiner notes that Cochrane et al. is relied upon to teach "the translation of a GROUP BY clause with the GROUPING SETS, ROLLUP, or CUBE operations into the plurality of levels, wherein each of the levels has one or more grouping sets comprised of grouping columns". It is also noted that the memo data structure of Al-omari et al. teaches "a search data structure used by the optimizer for representing elements of the search space. The Memo data structure is organized into equivalence classes denoted as groups. Each group includes one or more logical and physical expressions that are semantically equivalent to one another. Expressions are semantically equivalent if they produce the identical output. Initially each logical expression of the input query tree is represented as a separate group in memo". Cochrane et al. teaches multiple expressions in a query, and Al-omari et al. teaches to organize expressions into a data structure with an array of pointers, each pointer pointing to expressions for different levels. Thus, the claimed elements are taught by Cheng et al. in view of Cochrane et al., in view of Al-omari et al.

Appellants' attorney disagrees with this analysis.

The combination of Cheng, Cochrane and Al-omari does not teach or suggest all the limitations recited in Appellants' independent claim 1. Consider the cited portions of Cheng, Cochrane and Al-omari, which are set forth below:

> Cheng: Abstract
>
> In the early 1980's, researchers recognized that semantic information stored in databases as integrity constraints could be used for query optimization. A new set of techniques called semantic query optimization (SQO) was developed. Some of the ideas developed for SQO have been used commercially, but to the best of our knowledge, no extensive implementations of SQO exist today.
>
> In this paper, we describe an implementation of two SQO techniques, Predicate Introduction and Join Elimination, in DB2 Universal Database. We present the implemented algorithms and performance results using the TPCD and APB-1 OLAP benchmarks. Our experiments show that SQO can lead to dramatic query performance improvements. A crucial aspect of our implementation of SQO is the fact that it does not rely on complex integrity constraints (as many previous SQO techniques did); we use only referential integrity constraints and check constraints.
>
> Cheng: Page 2, Example 1
>
> Example 1. Consider the following two queries (both asked against the TPCD [19]). The first query illustrates the technique of Join Elimination.

```
Q1:   select      p_name, p_retailprice, s_name, s_address
      from        tpcd.lineitem, tpcd.partsupp, tpcd.part, tpcd.supplier
      where       p_partkey = ps_partkey and
                  s_suppkey = ps_suppkey and
                  ps_partkey = l_partkey and
                  ps_suppkey = l_suppkey and
                  l_shipdate between '1994-01-01' and
                  '1996-06-30' and l_discount > 0.1
      group by    p_name, p_retailprice, s_name, s_address
      order by    p_name, s_name;
```

> Cheng: Page 5, Query 1
>
> Let the query be Q1 of Example 1. The graphs describing the structure of the joins of the query are shown in Figure 1.
>
> <graphs>
>
> Thus, Q, can be optimized into Q'.

```
Q' :  select     p_name, p_retailprice, s_name, s_address
      from       tpcd.lineitem, tpcd.part, tpcd.supplier
      where      p_partkey = l_partkey and
                 s_suppkey = l_suppkey and
                 l_shipdate between '1994-01-01' and
                 '1996-06-30' and l_discount > 0.1
      group by   p_name, p_retailprice, s_name, s_address
      order by   p_name, s_name;
```

<u>Cochrane: Col. 7, Lines 26-30</u>

Generally, the query parser 92 lexes, parses, and semantically checks a query, producing an internal representation (a "query graph model") that is rewritten and submitted to the optimizer which generates an optimized query execution plan.

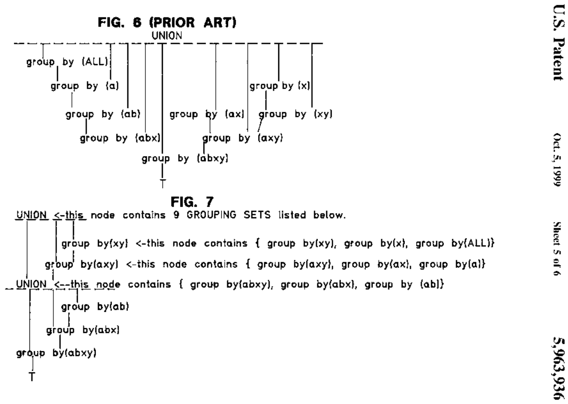
<u>Cochrane: Col. 7, Lines 44-48</u>

The system of FIG. 5 employs the invention to produce a QGM in which the number of GROUP BYs necessary to execute a GROUP BY with multiple GROUPING SETS, concatenated ROLLUPs, or a CUBE has been reduced.

<u>Cochrane: Col. 8, lines 26-42</u>

Now, utilizing the principles of the present invention, and noting the previously derived intersection results shown above at (1)-(4), it becomes possible to construct a query graph model that includes a stacking of GROUP BYs that results in the computation and planning of only 5 GROUP BYs as opposed to the 9 required in FIG. 6. This query graph model is shown in FIG. 7. It should be emphasized that the query graph model of FIG. 7 produces results that are identical to the solution provided in FIG. 6, with only 5 GROUP BY operations, a considerable economy in computational overhead. Indeed, this reduction in the number of GROUP BYs may, in an RDBMS implementing large multi-dimensional tables and subject to complex OLAP queries, be necessary to implement the query. This is due to the fact that the size of such queries, combined with the prior art, can require such large-scale computational assets as to render the query incapable of implementation.

Cochrane: FIGS. 6 and 7



Cochrane: Col. 11, line 62 – col. 12, line 15

As an example, consider the following: GROUP BY ROLLUP(a,b), ROLLUP(x,y) in which the GROUP BY’s for ROLLUP(a,b) are:

GROUP BY(a,x,y)

GROUP BY(x,y)

and the GROUP BY’s for ROLLUP(x,y) are:

GROUP BY(a,b,x)

GROUP BY(a,b)

Now, the base group for ROLLUP(a,b)ROLLUP(x,y) is determined by base step:

```
UNION1
|||
||GROUP BY(x,y)
|||
|GROUP BY(a,x,y)
||
GROUP BY(a,b,x,y)
|
T
```

Al-omari: FIGS. 3C and 3D

U.S. Patent Aug. 20, 2002 Sheet 4 of 25 US 6,438,741 B1

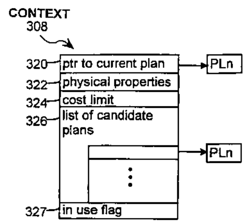


FIG. 3C

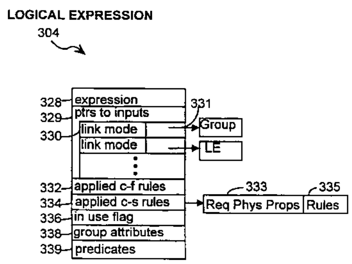


FIG. 3D

Al-omari: col. 10, lines 36-48

Memo: A memo is a search data structure used by the optimizer for representing elements of the search space. The Memo is organized into equivalence classes denoted as groups. Each group includes one or more logical and physical expressions that are semantically equivalent to one another. Expressions are semantically equivalent if they produce the identical output. Initially each logical expression of the input query tree is represented as a separate group in Memo. As the optimizer applies rules to the expressions in the groups, additional equivalent expressions and groups are added. Each group also contains one or more plans and contexts. A context represents plans having the same optimization goal.

Al-omari: col. 14, lines 26-35 and lines 41-43 (actually lines 26-50)

Referring to FIGS. 3A-3E, the Memo 122 includes one or more groups 302, where each group 302 contains an array of pointers to one or more logical expressions 304, an array of pointers to one or more physical expressions 306, an array of pointers to one or more contexts 308, an array of pointers to one or more plans 305, and an exploration pass indicator 307. A logical expression, physical expression, context, and plan are described in more detail below. An exploration pass indicator 307 indicates for each pass whether or not the group has been explored. Preferably, the exploration pass indicator is a bitmap having n bits with

one or more bits representing a particular pass and indicating whether or not exploration was performed in the pass.

Each logical expression 304 is represented as a data structure that stores the particular expression 328 and has pointers 331 associated with each input expression 329. Each pointer 331 has a link mode 330 that specifics the datum that the pointer addresses. Preferably, there are two link modes associated with an input expression: a memo mode and a binding mode. In memo mode, the pointer 331 identifies the group corresponding to the input expression. In binding mode, the pointer 331 identifies a logical expression that is part of a binding.

Al-omari: col. 3, lines 18-19

The invention is a system and method for optimizing complex SQL database queries.

Appellants' attorney respectfully submits that the combination of Cheng, Cochrane and Al-omari does not teach or suggest the limitations "during compilation of the query, maintaining a GROUP BY clause with one or more GROUPING SETS, ROLLUP or CUBE operations in its original form, instead of rewriting the GROUP BY clause, until after query rewrite," and "at a later stage of query compilation, translating the GROUP BY clause with the GROUPING SETS, ROLLUP or CUBE operations into a plurality of levels, wherein each of the levels has one or more grouping sets comprised of grouping columns, and generating a query execution plan for the query with a super group block having an array of pointers, wherein each pointer points to the grouping sets for a particular one of the levels."

For example, the query of Cheng merely illustrates the technique of join elimination, in the context of semantic query optimization (SQO). However, Cheng merely shows the GROUP BY clause in the same form in both the original query and the optimized query, indicating that the GROUP BY clause is not "maintained during compilation" and then "translated at a later stage of query compilation," as recited in Appellants' claims. Instead, the GROUP BY clause of Cheng is apparently left untouched during the join elimination optimization. Moreover, the GROUP BY clause in Cheng does not have GROUPING SETS, ROLLUP or CUBE operations in its original form, as admitted by the Office Action. Therefore, the query of Cheng, and the example cited by the Office Action, have no relevance to Appellants' claims.

In another example, the description from Cochrane set forth above merely describes the translation of a query into a "query graph model" that is rewritten and submitted to an optimizer

which generates an optimized query execution plan, wherein the optimization of GROUP BYs is performed by stacking, which reduces the number of GROUP BYs while producing identical results. However, this optimization scheme of Cochrane says nothing about maintaining a GROUP BY clause with one or more GROUPING SETS, ROLLUP or CUBE operations in its original form, instead of rewriting the GROUP BY clause, until after query rewrite. Instead, the optimization scheme of Cochrane reduces the GROUP BYs during query rewrite, which means that the GROUP BY clause is not maintained in its original form until after query rewrite, but instead the GROUP BY clause is rewritten.

In yet another example, the description from Al-omari set forth above describes a memo structure that is organized into equivalence classes denoted as groups, wherein each group includes one or more logical and physical expressions that are semantically equivalent to one another in that they produce an identical output. However, the groups in Al-omari are in no way equivalent to Appellants' claimed super group block. Specifically, the memo structure of Al-omari includes one or more groups, where each group contains an array of pointers to one or more logical expressions, an array of pointers to one or more physical expressions, an array of pointers to one or more contexts, an array of pointers to one or more plans, and an exploration pass indicator. In Appellants' claims, on the other hand, the super group block supports the translation of a GROUP BY clause with the GROUPING SETS, ROLLUP or CUBE operations into a plurality of levels, wherein each of the levels has one or more grouping sets comprised of grouping columns, the super group block has an array of pointers, and each pointer of the super group block points to the grouping sets for a particular one of the levels. This super group block of Appellants' claims recites different structure and functions as compared to the memo structure of Al-omari.

Thus, Appellants' attorney submits that independent claim 1 is allowable over the combination of Cheng, Cochrane and Al-omari. Further, dependent claims 2 and 3 are submitted to be allowable over the combination of Cheng, Cochrane and Al-omari in the same manner, because they are dependent on independent claim 1, and thus contain all the limitations of the independent claim. In addition, dependent claims 2 and 3 recite additional novel elements not shown by the combination of Cheng, Cochrane and Al-omari, as set forth below.

2. Dependent Claim 2

Appellants' invention, as set forth in dependent claim 2, which is dependent on claim 1, recites that the method of claim 1 further comprises: (1) at query execution time, dynamically determining a grouping sets sequence for the GROUP BY clause with the GROUPING SETS, ROLLUP or CUBE operations based on intermediate grouping sets, in order to optimize the grouping sets sequence. The Office Action asserts that these limitations are shown by Cochrane at col. 8:26-42 and Figure 7. At the indicated locations, Cochrane merely describes constructing a query graph model (QGM) that includes a stacking of GROUP BYs. However, Cochrane says nothing about group sets sequences, or dynamically determining, at query execution time, a grouping sets sequence for a GROUP BY clause with GROUPING SETS, ROLLUP or CUBE operations, based on intermediate grouping sets, in order to optimize the grouping sets sequence.

3. Dependent Claim 3

Appellants' invention, as set forth in dependent claim 3, which is dependent on claim 2, recites that the dynamically determining step of claim 2 further comprises (1) performing a GROUP BY for a base grouping set and then optimizing execution of the grouping sets sequence by selecting a grouping set having lowest cardinality from a previous one of the levels as an input to a grouping set on a next one of the levels, and (2) performing a UNION ALL operation on the grouping sets. The Office Action asserts that these limitations are shown by Cochrane at col. 11:47-49 and Figure 7. At the indicated locations, Cochrane merely describes the base step in the construction of a stack of GROUP BYs for a ROLLUP, wherein a base GROUP BY and all other GROUP BYs for the ROLLUP are "unioned" together. However, Cochrane says nothing about grouping sets sequences, or performing a GROUP BY for a base grouping set and then optimizing execution of the grouping sets sequence by selecting a grouping set having lowest cardinality from a previous one of the levels as an input to a grouping set on a next one of the levels, and then performing a UNION ALL operation on the grouping sets.

G. Summary.

The references, taken individually or in combination, fail to teach the Appellants' claimed invention. Further, the various elements of the Appellants' claimed invention together provide operational advantages over the systems disclosed in the references. In addition, Appellants' invention solves problems not recognized by the references. Consequently, Appellants' attorney submits that claims 1-3 are allowable over the references.

VIII. CONCLUSION

In light of the above arguments, Appellants' attorney respectfully submits that the cited references do not anticipate nor render obvious the claimed invention. More specifically, Appellants' claims recite novel physical features which patentably distinguish over any and all references under 35 U.S.C. §§ 102 and 103.

As a result, a decision by the Board of Patent Appeals and Interferences reversing the Examiner and directing allowance of the pending claims in the subject application is respectfully solicited.

Respectfully submitted,

GATES & COOPER LLP
Attorneys for Appellants

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: December 18, 2008

By: /George H. Gates/
Name: George H. Gates
Reg. No.: 33,500

GHG/

CLAIMS APPENDIX

1. A method of optimizing a query in a computer system, the query being performed by the computer system to retrieve data from a database stored on the computer system, the method comprising:

(a) during compilation of the query, maintaining a GROUP BY clause with one or more GROUPING SETS, ROLLUP or CUBE operations in its original form, instead of rewriting the GROUP BY clause, until after query rewrite;

(b) at a later stage of query compilation, translating the GROUP BY clause with the GROUPING SETS, ROLLUP or CUBE operations into a plurality of levels, wherein each of the levels has one or more grouping sets comprised of grouping columns, and generating a query execution plan for the query with a super group block having an array of pointers, wherein each pointer points to the grouping sets for a particular one of the levels; and

(c) performing the query execution plan to retrieve data from a database stored on the computer system.

2. The method of claim 1, further comprising:

(1) at query execution time, dynamically determining a grouping sets sequence for the GROUP BY clause with the GROUPING SETS, ROLLUP or CUBE operations based on intermediate grouping sets, in order to optimize the grouping sets sequence.

3. The method of claim 2, wherein the dynamically determining step further comprises (1) performing a GROUP BY for a base grouping set and then optimizing execution of the grouping sets sequence by selecting a grouping set having lowest cardinality from a previous one of the levels as an input to a grouping set on a next one of the levels, and (2) performing a UNION ALL operation on the grouping sets.

4-9. (CANCELED)

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.